

WHAT IS CLAIMED IS:

1. A computer readable medium containing computer executable instructions to perform a method for assisting a computer programmer in real time to complete a programming language statement in a computer program, said method comprising:

enabling a programming language editor having a character position cursor and a randomly positionable pointer;

partially compiling available ones of a plurality of programming language statements in said computer program;

defining a finite set of programming language statement information that is relevant to at least one segment of a present programming language statement from among said plurality of programming language statements that is proximate to said character position cursor; and

15 generating a passive assist window that contains said finite set of programming language statement information in a location proximate to said character position cursor.

2. A method according to claim 1 including:

20 automatically attempting said steps of claim 1 for each character received by said programming language editor.

3. A method according to claim 2 including:

attempting said steps of claim 1 on a randomly selected one of said 25 plurality of programming language statements in response to a real time request by said computer programmer.

4. A method according to claim 1 including:

attempting said steps of claim 1 on a randomly selected one of said 30 plurality of programming language statements in response to a real time request by said computer programmer at a time when at least one automatic assist window feature is disabled.

5. A method according to claim 1 including:  
generating a simultaneous plurality of passive assist windows that each contain a finite set of programming language statement information in a location proximate to said character position cursor, said simultaneous plurality of 5 passive assist windows being selected from at least one of a group comprised of: a selection menu assist window and an informational display assist window.

6. A method according to claim 1 wherein said step of defining includes:  
10 generating said finite list as a selectable list of menu items that can each validly complete said at least one segment of said present programming language statement that is proximate to said character position cursor.

7. A method according to claim 1 wherein said step of generating 15 includes:  
creating a selection menu assist window comprised of said list of menu items; and  
enabling window control features for said selection menu assist window.

20 8. A method according to claim 6 including:  
replacing said at least one segment of said present programming language statement with one of said list of menu items in response to an input command by said computer programmer.

25 9. A method according to claim 1 wherein said step of defining includes:  
generating an argument list of each argument in said present programming language statement; and  
identifying an argument type for each argument in said argument list 30 selected from at least one of a group comprised of: a mandatory argument and an optional argument.

10. A method according to claim 9 wherein said step of generating

includes:

reverse parsing said present programming language statement into a plurality of tokens that each represent an individual component selected from at least one of a group comprised of: an object entity segment and a delimiter, in 5 response to a real time request by said computer programmer;

distinguishing said plurality of tokens between a procedure call token and any argument token in said argument list; and

binding said argument list.

10 11. A method according to claim 1 wherein said step of generating includes:

generating an informational display assist window based on an argument list;

15 distinguishing a mandatory argument from an optional argument within said argument list; and

highlighting a present argument within said argument list that corresponds to a present location of said character position cursor within said present programming language statement.

20 12. A system for passively assisting a user in real time to complete a programming language statement, said system comprising:

a programming language editor having a character position cursor and a randomly positionable pointer;

means for partially compiling available ones of a plurality of programming 25 language statements in said computer program; and

means for generating an assist window that contains a finite set of programming language statement information in a location proximate to said character position cursor, said assist window being selected from at least one of a group comprised of: a selection menu assist window and an informational

30 display assist window.

13. A system according to claim 12 wherein said means for generating includes:

means for identifying a desired menu item from said selection menu assist window; and

means for replacing a segment of a present programming language statement at a present location of said character position cursor with said

5 desired menu item in response to said means for identifying.

14. A system according to claim 12 including:

means for displaying information in an informational display assist window, said information being related to at least one segment of a present programming language statement that is proximate a present location of said character position cursor and selected from at least one of a group comprised of: a symbol definition, a defined constant, a procedure call map, and an enumerated list.

15. A system according to claim 12 including:

means for automatically enabling said means of claim 12 for each character received by said programming language editor.

16. A system according to claim 12 including:

20 means for enabling said means of claim 1 on a randomly selected one of  
said plurality of programming language statements in response to a real time  
request by said user and independent of any automatic assist feature.

17. A real time method for assisting a user to complete a programming language statement in a computer program, said real time method comprising:

enabling a programming language editor having a character position cursor;

continuously resolving symbolic portions of available ones of a plurality of programming language statements into a partial program compilation;

30 identifying a present programming language statement and at least one segment of said present programming language statement based on a location of said character position cursor;

determining a finite set of information related to said present

programming language statement and said at least one segment of said present programming language statement based on said partial program compilation; and

generating an assist window of said finite set of information.

5

18. A method according to claim 17 wherein said step of identifying includes:

determining an identity of input to said programming language editor by said user;

10 enabling a reverse parse evaluation of said present programming language statement into identifiable tokens for each of said at least one segment therein in response to said input being an on-demand request by said user;

enabling execution of a editing task in response to said input being a programming language editor command;

15 enabling a first type of commit of an identified menu item from a selection menu assist window in response to said input being a commit key, wherein said step of enabling a first type of commit includes:

identifying said commit key as a non-delimiter type commit key; and

20 discarding said commit key;

enabling a second type of commit of an identified menu item from a selection menu assist window in response to said input being a commit key, wherein said second type of commit includes:

identifying said commit key as a delimiter type commit key; and

25 inserting said commit key after said identified menu item in said present programming language statement; and

adding to said present programming language statement at a location of said character position cursor in response to said input being a non-commit key type input character.

30

19. A method according to claim 17 wherein said step of generating includes:

displaying a selection menu assist window where said present

programming language statement is identified as an operator embedded programming language statement; and

displaying an informational display assist window where said present programming language statement is identified as a non-operator embedded 5 programming language statement.

20. A method according to claim 19 wherein said non-operator embedded programming language statement is a procedure call.

add  
a2  
ADD B<sup>3</sup> /